Computer Security

Mini Project - RC4 Cipher (WEB)

Submitted by:

Mohamed Abd EL-Raouf Mohamed (G6)

Presented To:

Prof. Doc.Mahmoud ElShishtawy

Dr. Abdelrahman Essam

Chapter 1: Introduction

RC4 is a stream cipher designed in 1987 by Ron Rivest for RSA Security. It is a variable key-size stream cipher with byte-oriented operations. The algorithm is based on the use of a random permutation. Analysis shows that the period of the cipher is overwhelmingly likely to be greater than 10^100. Eight to sixteen machine operations are required per output byte, and the cipher can be expected to run very quickly in software. RC4 is probably the most widely used stream cipher. It is used in the SSL/TLS secure web protocol, & in the WEP & WPA wireless LAN security protocols. RC4 was kept as a trade secret by RSA Security, but in September 1994 was anonymously posted on the Internet on the Cypher punks anonymous remailers list. In brief, the RC4 key is used to form a random permutation of all 8-bit values, it then uses that permutation to scramble input info processed a byte at a time.

Chapter 2 : How It Work

2.1 Description

- stream cipher symmetric key
- Use two array, state and key

1. 256-byte state table.
   State[256]=[0 .. 255]
2. It has the capability of using keys between 1 and 2048 bits.
   Key[1..2048] = […….]

Hint. WEP use 40 bits

2.2 Algorithm

Two phases

- Key Setup

  1. f = (f + Si + Kg) mod 4

  2. Swapping Si with Sf

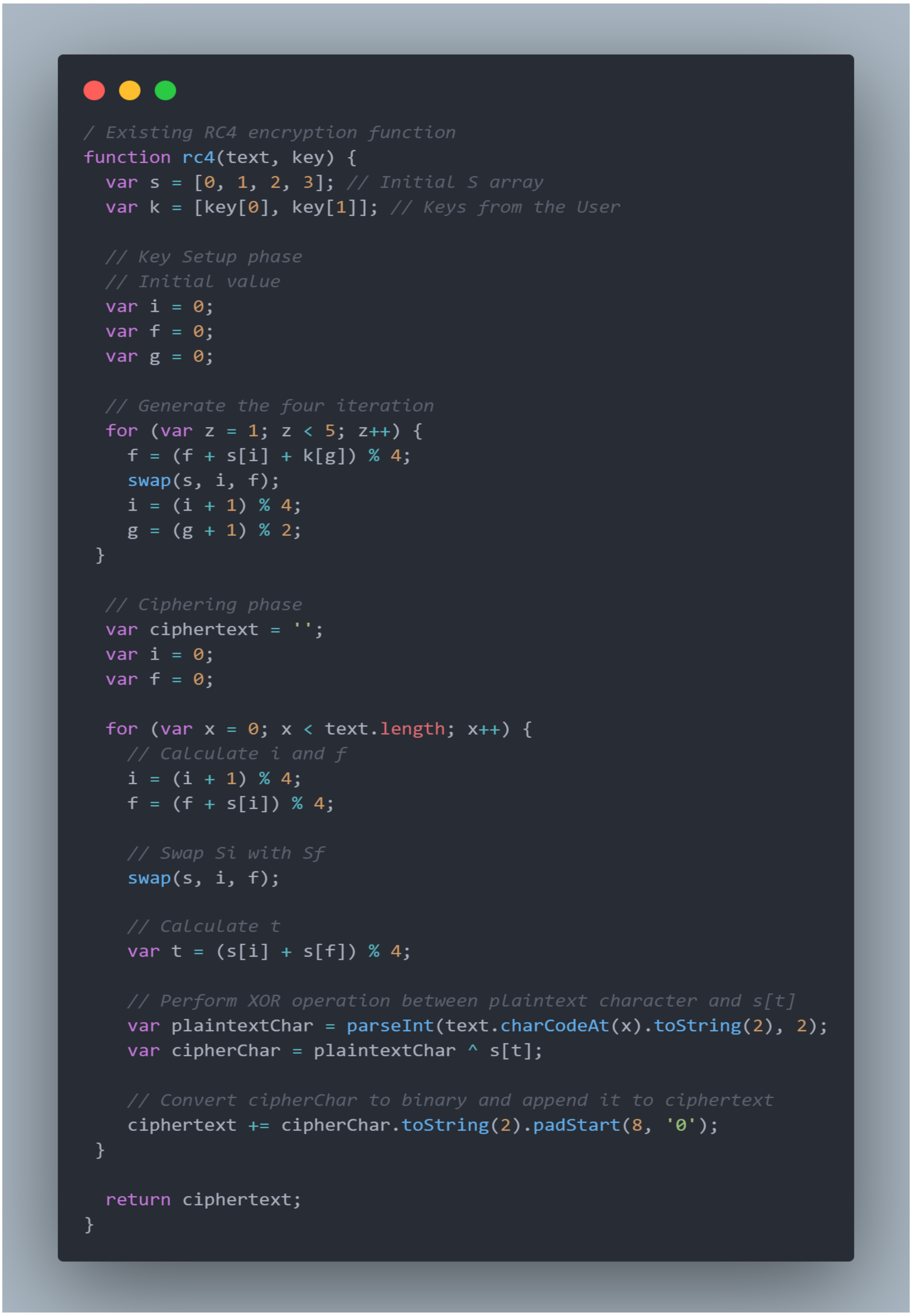
- Ciphering (XOR)

  1. i = (i + 1) mod 4 , and f = (f + Si) mod 4

  2. Swaping Si with Sf

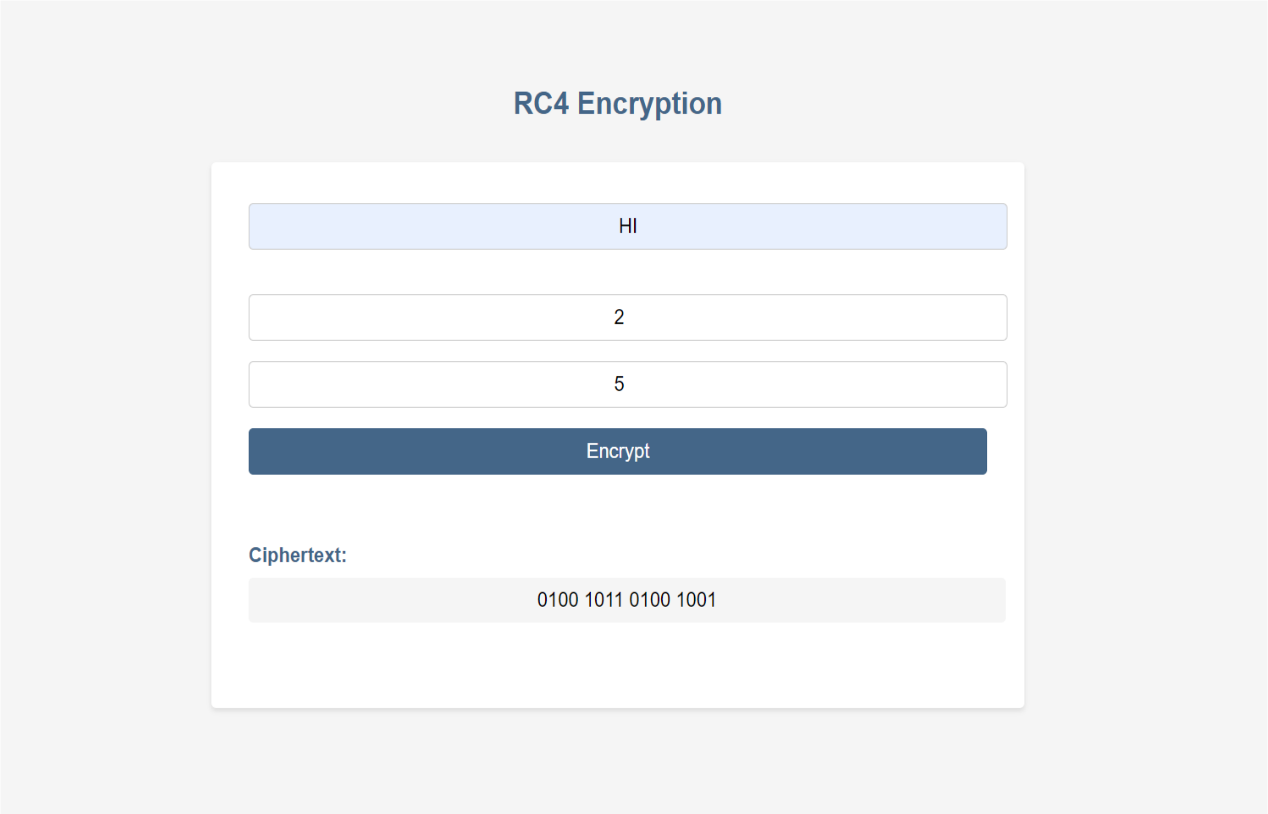
  3. t = (Si + Sf) mod 4

Chapter 3: Implementation



```
/ Existing RC4 encryption function
function rc4(text, key) {
  var s = [0, 1, 2, 3]; // Initial S array
  var k = [key[0], key[1]]; // Keys from the User

  // Key Setup phase
  // Initial value
  var i = 0;
  var f = 0;
  var g = 0;

  // Generate the four iteration
  for (var z = 1; z < 5; z++) {
    f = (f + s[i] + k[g]) % 4;
    swap(s, i, f);
    i = (i + 1) % 4;
    g = (g + 1) % 2;
 }

  // Ciphering phase
  var ciphertext = '';
  var i = 0;
  var f = 0;

  for (var x = 0; x < text.length; x++) {
    // Calculate i and f
    i = (i + 1) % 4;
    f = (f + s[i]) % 4;

    // Swap Si with Sf
    swap(s, i, f);

    // Calculate t
    var t = (s[i] + s[f]) % 4;

    // Perform XOR operation between plaintext character and s[t]
    var plaintextChar = parseInt(text.charCodeAt(x).toString(2), 2);
    var cipherChar = plaintextChar ^ s[t];

    // Convert cipherChar to binary and append it to ciphertext
    ciphertext += cipherChar.toString(2).padStart(8, '0');
 }

  return ciphertext;
}
```

GUI



RC4 Encryption

HI

2

5

Ciphertext:

0100 1011 0100 1001